

THE switch-case STATEMENT

```
.....  MATLAB program.
.....

switch switch expression
  case value1
.....   ] Group 1 of commands.
.....   ]
  case value2
.....   ] Group 2 of commands.
.....   ]
  case value3
.....   ] Group 3 of commands.
.....   ]
  otherwise
.....   ] Group 4 of commands.
.....   ]
end
.....  MATLAB program.
.....
```

Figure 6-4: The structure of a switch-case statement.

Example 1:- Write a program to find the value of y such that

$$y = \begin{cases} x^2 - 6 & \text{if } x = -2 \\ 7x^3 & \text{if } x = 2 \\ x^2 - 5x - 1 & \text{if } x = 8 \\ 9x - 4 & \forall x / \{-2, 2, 8\} \text{ or otherwise} \end{cases}$$

```
x=input('x=');
switch x
  case -2
    y=x^2-6
  case 2
    y=7*x^3
  case 8
    y=x^2+5*x-1
  otherwise
```

$$y=9*x-4$$

end

Example 2:- Write a program to find the value of y such that

$$y = \begin{cases} x^4 - x^3 + 7 & \text{if } x = 2 \\ \cos x & \text{if } x = 5 \\ \sin x & \text{if } x = -7 \\ \tan x & \text{otherwise} \end{cases}$$

```
x=input('x=');
switch x
    case 2
        y=x^4-3*x^3+7;
        disp(y)
    case 5
        y=cos(x);
        disp(y)
    case -7
        y=sin(x)-1;
        disp(y)
    otherwise
        y=tan(x);
        disp(y)
end
```

LOOPS

A loop is another method to alter the flow of a computer program. In a loop, the execution of a command, or a group of commands, is repeated several times consecutively. Each round of execution is called a pass. In each pass at least one variable, but usually more than one, or even all the variables that are defined within the loop

MATLAB has two kinds of loops. 1- **for-end loops**, 2- **while-end loops**

for-end Loops

In *for-end* loops the execution of a command, or a group of commands, is repeated a predetermined number of times. The form of a loop is shown in Figure 6-5.

- The loop index variable can have any variable name (usually *i*, *j*, *k*, *m*, and *n* are used, but *i* and *j* should not be used if MATLAB is used with complex numbers).

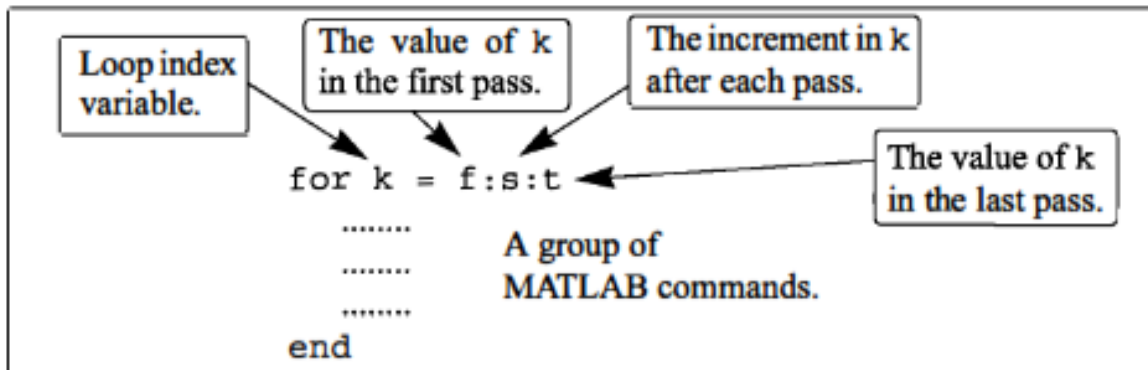


Figure 6-5: The structure of a for-end loop.

A simple example of a for-end loop (in a script file) is:

```
for k=1:3:10
    x = k^2
end
```

When this program is executed, the loop is executed four times. The value of k in the four passes is $k = 1, 4, 7,$ and 10 , which means that the values that are assigned to x in the passes are $x = 1, 16, 49,$ and 100 , respectively. Since a semicolon is not typed at the end of the second line, the value of x is displayed in the Command Window at each pass. When the script file is executed, the display in the Command Window is:

```
>> x =
     1
x =
    16
x =
    49
x =
   100
```

Example 1:- Write a program to find $\sum_{i=1}^{50} i$

`%s:sum of number ,`

```
s=0;
for i=1:50
    s=s+i;
end
disp(s);
```

Example 2:- Write a program to find

- 1- The summation of even number from 1 to 20
- 2- The average of them.

```
%se:sum of even number ,ne: number of even numbers
```

```
s=0;ne=0;
for i=2:2:20
    s=s+i;
    ne=ne+1;
end
avr=s/ne;
disp(s);
disp(avr);
```

(H.W)

Q: - Write a program to find

- 1) The summation of odd number from 1 to 20
- 2) The average of them.

Example 3:- Write a program to find the summation of even number and the average of odd numbers from 1 to 300 ?

```
se=0;so=0;ne=0;no=0;
for i=1:300
    if rem(i,2)==0
        se=se+i;
    else
        so=so+i;
        no=no+1;
    end
end
avo=s/no;
disp(se);
```

```
disp(avo);
```

Example 4:- Write a program to input (n) number then find

- 1- The summation
- 2- The factorial

Of the prime numbers among them ?

```
n=input('n=');
sp=0;
for i=1:n
    if isprime(i)
        sp=sp+i;
        f=factorial(i);
        disp('      n      f      ');
        disp([i, f])
    end
end
disp('sum of prime numbers')
disp(sp)
```

Example 5:- Use a for-end loop in script file to calculate the sum of the first

n terms of series $\sum_{k=1}^n \frac{(-1)^k k}{2^k}$

```
n=input('n=');
s=0;
for k=1:n
    s=s+(-1)^k*k/2^k;
end
disp('the sum of the series is');
disp(s);
```

Example 6:- Use a for-end loop in script file to calculate the sum of the first n terms of series $\sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!}$, Execute the script file for $n = 4, x = 2$.

%Where n is number of terms in the series.

```
n=input('n=');
x=input('x=');
s=0;
for k=1:n
    s=s+(-1)^k*x^(2*k+1)/factorial(2*k+1);
end
disp('the sum of the series is');
disp(s);
```

Home Work

Write program to find

$$1) \sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + \frac{x^n}{n!}$$

$$2) z = \frac{(x+y)}{3!} + \frac{(x+y)^3}{5!} + \frac{(x+y)^5}{7!} + \dots + \frac{(x+y)^{27}}{29!}$$

$$3) \cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + \frac{x^n}{n!}$$

4- A vector is given by

$V = [5, 17, -3, 8, 0, -7, 12, 15, 20, -6, 6, 4, -7, 16]$. Write a program as a script file that doubles the elements that are positive and are divisible by 3 or 5, and, raises to the power of 3 the elements that are negative but greater than -5.

NESTED LOOPS AND NESTED CONDITIONAL STATEMENTS

Loops and conditional statements can be nested within other loops or conditional statements. This means that a loop and/or a conditional statement can start (and end) within another loop or conditional statement. There is no limit to the number of loops and conditional statements that can be nested. It must be remembered, however, that each `if`, `case`, `for`, and `while` statement must have a corresponding end statement. Figure 6-7 shows the structure of a nested `for-end`

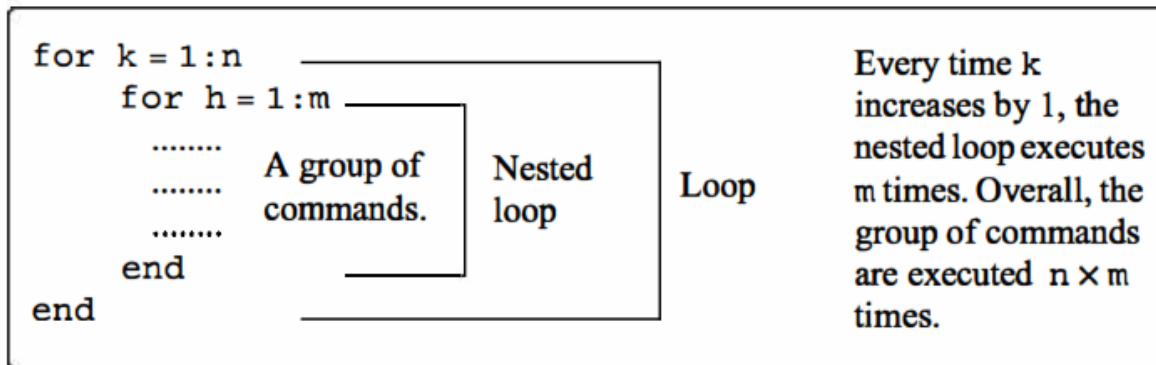


Figure 6-7: Structure of nested loops.

Sample Problem 6-8: Creating a matrix with a loop

Write a program in a script file that creates an $n \times m$ matrix with elements that have the following values. The value of each element in the first row is the number of the column. The value of each element in the first column is the number of the row. The rest of the elements each has a value equal to the sum of the element above it and the element to the left. When executed, the program asks the user to enter values for n and m .

Solution

The program, shown below, has two loops (one nested) and a nested `if-elseif-else-end` structure. The elements in the matrix are assigned values row by row. The loop index variable of the first loop, k , is the address of the row, and the loop index variable of the second loop, h , is the address of the column.

```
n=input('Enter the number of rows ');
m=input('Enter the number of columns ');
A=[];
for k=1:n
    for h=1:m
        if k==1
            A(k,h)=h;
```

Define an empty matrix A

Start of the first for-end loop.

Start of the second for-end loop.

Start of the conditional statement.

Assign values to the elements of the first row.


```

elseif h==1
    A(k,h)=k;
else
    A(k,h)=A(k,h-1)+A(k-1,h);
end
end
end
end
A

```

Assign values to the elements of the first column.

Assign values to other elements.

end of the if statement.

end of the nested for-end loop.

end of the first for-end loop.

The program is executed in the Command Window to create a 4×5 matrix.

```

A =
     1     2     3     4     5
     2     4     7    11    16
     3     7    14    25    41
     4    11    25    50    91

```

Example 2:- Write a program in script file to construct a matrix A which the value of each element in the matrix A equal to summation of the number of column and row which contain this element

```

n=input('n=');
m=input('m=');
for i=1:n
for j=1:m
    A(i,j)=i+j;
end
end
A

```

Quiz

Write a program in script file to construct a matrix A as following

$$A = \begin{bmatrix} 9 & 8 & 8 \\ 11 & 9 & 8 \\ 11 & 11 & 9 \end{bmatrix}$$

Home Work

hidden solution

A symmetric (5×5) Pascal matrix is displayed on the right. Write a MATLAB program that creates an $n \times n$ symmetric Pascal matrix. Use the program to create 4×4 and 7×7 Pascal matrices.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 6 & 10 & 15 \\ 1 & 4 & 10 & 20 & 35 \\ 1 & 5 & 15 & 35 & 70 \end{bmatrix}$$

while-end *Loops*

while-end loops are used in situations when looping is needed but the number of passes is not known in advance. In while-end loops the number of passes is not specified when the looping process starts. Instead, the looping process continues until a stated condition is satisfied. The structure of a while-end loop is shown in Figure 6-6.

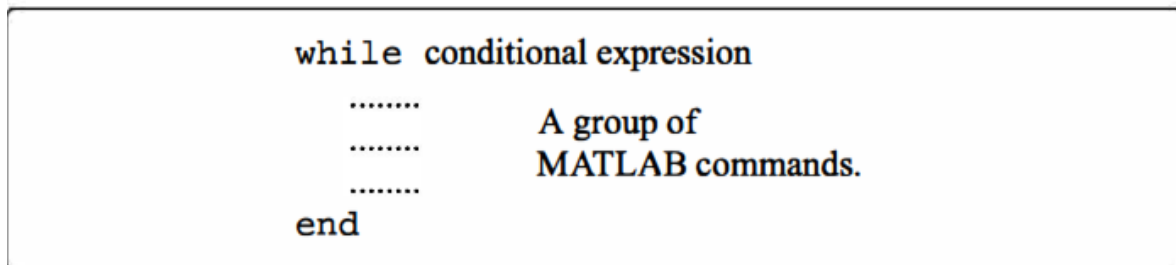


Figure 6-6: The structure of a while-end loop.

this program a variable x with an initial value of 1 is doubled in each pass as long as its value is equal to or smaller than 15.

```
x=1
while x<=15
    x=2*x
end
```

Initial value of x is 1.

The next command is executed only if $x \leq 15$.

In each pass x doubles.

When this program is executed the display in the Command Window is:

```
x =
    1
x =
    2
x =
    4
x =
    8
x =
   16
```

Initial value of x .

In each pass x doubles.

When $x = 16$, the conditional expression in the `while` command is false and the looping stops.

Example 1: Write program to find the summation of number from 1 to 400

```
s=0;
i=1;
while i<=400
    s=s+i;
end
disp(s);
```

Example 2: Write program to find the summation $\sum_{i=1}^n i$ which not more than 300

```
s=0;
i=1;
while s<=300
    s=s+i;
    i=i+1;
end
```

```
disp(s-i);
```

Example 3: Write program to find the summation of even numbers between 50 and 250 by using 1- **for-end** 2- **while-end**

1-

```
se=0;
i=1;
for i=50:2:250
    se=se+i;
end
disp(se)
```

2-

```
se=0;
i=50;
while i<=250
    if rem(i,2)==0
        se=se+i;
    end
end
disp(se)
```

Quiz

Q/ Write program to input matrix **A** to know that **A** is symmetric or not?

Home Work

Q/ Write program to find the summation where

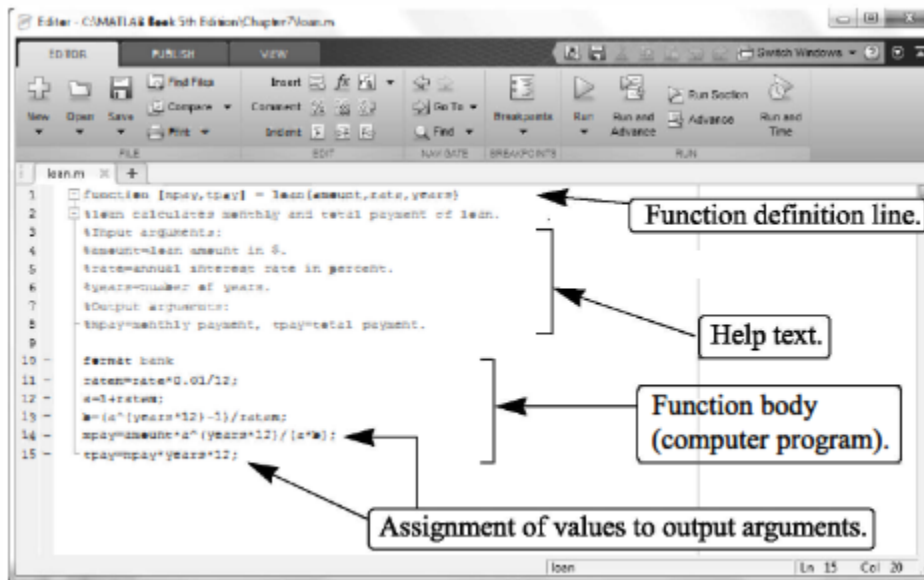
$$S = \sqrt{1^5} + \sqrt{3^5} + \sqrt{5^5} + \dots + \sqrt{n^5}$$

Find the largest **n** st. $S \leq 90$

Function m-file

Functions are m-file that can accept input argument.

STRUCTURE OF A FUNCTION FILE

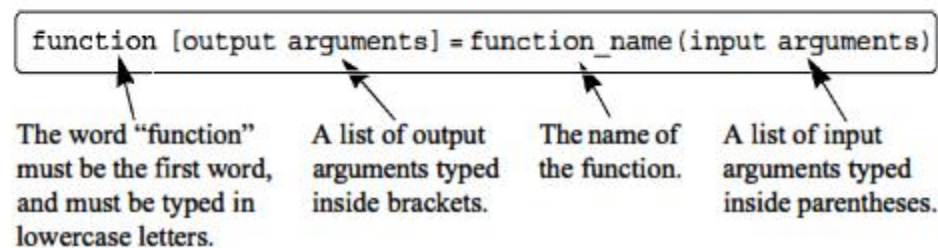


Function Definition Line

The first executable line in a function file *must* be the function definition line. Otherwise the file is considered a script file. The function definition line:

- Defines the file as a function file
- Defines the name of the function
- Defines the number and order of the input and output arguments

The form of the function definition line is:



Function definition line

Comments

function [mpay,tpay] = loan(amount,rate,years)	Three input arguments, two output arguments.
function [A] = RectArea(a,b)	Two input arguments, one output argument.
function A = RectArea(a,b)	Same as above; one output argument can be typed without the brackets.
function [V, S] = SphereVolArea(r)	One input variable, two output variables.
function trajectory(v,h,g)	Three input arguments, no output arguments.

Function definition line

File name

function [mpay,tpay] = loan(amount,rate,years)	loan.m
function [A] = RectArea(a,b)	RectArea.m
function [V, S] = SphereVolArea(r)	SphereVolArea.m
function trajectory(v,h,g)	trajectory.m

Sample Problem 7.1 : User-defined function for a math function

Write a function file (name it `chp7one`) for the function $f(x) = \frac{x^4 \sqrt{3x+5}}{(x^2+1)^2}$. The input to the function is x and the output is $f(x)$. Write the function such that x can be a vector. Use the function to calculate:

- (a) $f(x)$ for $x = 6$.
- (b) $f(x)$ for $x = 1, 3, 5, 7, 9$, and 11 .

Solution

The function file for the function $f(x)$ is:

```
function y=chp7one(x)
y=(x.^4.*sqrt(3*x+5))./(x.^2+1).^2;
```

Function definition line.
Assignment to output argument.

```
>> chp7one(6)
ans =
    4.5401
```

(b) To calculate the function for several values of x , a vector with the values of x is created and then used for the argument of the function.

```
>> x=1:2:11
x =
     1     3     5     7     9    11
```

```
>> chp7one(x)
ans =
     0.7071     3.0307     4.1347     4.8971     5.5197     6.0638
```

Example 1: Write program to find $f(x) = x^7 - 3x^4 + 2x - 11$ at $x = 5$, using function.

```
1- function y=f(x)
2- y=x^7-3*x^4+2*x-11;
3- return
```

```
% save file as (f)
```

```
>> f(5)
```

```
ans =
```

```
76249
```

Example 2: Write program to find $y_1 = 2 * x_1 - 3 * x_2$, $y_2 = 5 * x_1 + 3 * x_2$ at $x_1 = 5$, $x_2 = 6$ by using one m-function.

```
1-function [y1, y2]=f(x1, x2)
2-y1=2*x1-3*x2;
3-y2=5*x1+3*x2;
4-return
```

```
% save file as (f)
```

```
1-x1=input('x1=');
2-x2=input('x2=');
3-[y1, y2]=f(x1, x2);
```

```
4-disp([y1,y2]);  
Save file as any name you wont
```

```
>> Untitled2
```

```
x1=5
```

```
x2=6
```

```
    -8    43
```

Example 3: Write program to input (3) degree of a student to find the average (use function) then induce if the student pass or not ?

```
function y=avr(a,b,c)  
y=(a+b+c)/3;  
return  
  
% save file as (avr)  
a1=input('a=');  
a2=input('b=');  
a3=input('c=');  
z=avr(a1,a2,a3);  
disp(z);  
if z>=50  
    disp('the student is pass')  
else  
    disp('the student is faild')  
end
```



```
>> Untitled2
a=50
b=50
c=49
    49.6667
```

the student is faild

Example 4: Write program to input (n) real number (x) and find y by using function? Where $y = 9x^3 - 5x + 2$

```
function yy=f(xx)
yy=9*(xx)^3-5*(xx)+2;
return
```

```
% save file as (f)
```

```
n=input('n=');
for i=1:n
x=input('x=');
y=f(x);
disp('      n      x      y')
disp([i x y])
end
```

Example 5: By using function write program to find value of y ? Where

$$y = \begin{cases} \frac{\sin x + 1}{x} & x \neq 3, -3 \\ e^x - 2x & x = 3, -3 \end{cases}$$

```
function yy=f(xx)
switch xx
    case {3,-3}
        yy=exp(xx)-2*(xx);
```

```

        otherwise
        yy=(sin(xx)+1)/(xx);
end
    return

% save file as (f)

x=input('x=');
y=f(x);
disp(y);

```

Quiz

Q: Write program to find $n!$ by using function
 1-With using factorial function?
 2-Without using factorial function?

Home Work

Q/ Write program to find value of y by using function where

$$y = \sqrt{6\pi} \sum_{i=1}^n \frac{1}{i^{2i}}$$

In case i is even number, use
 $n = 20$

Q/ Write program to input the value of A, B, C to find the solution of equation in second degree $Ax^2 + Bx + C = 0$ in real number by using **if-elseif-else-end** statement?

Hint

To solve $ax^2 + bx + c = 0$:

1st: Use the square-root method if the x -term is missing.

2nd: Use the quadratic formula (QF).

3rd: **the equation has no solution in real number**



Solution is hidden